

ПРОГРАМНА СИСТЕМА ДЛЯ ПОШУКУ МЕДИКАМЕНТІВ У РЕЖИМІ ОНЛАЙН

Ю. В. Горбачов, І. І. Пастернак

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин
Емейли авторів: yura9875@ukr.net, pas_irusj@ukr.net

© Горбачов Ю.В., Пастернак І.І., 2021

У роботі наведено програмну систему для пошуку медикаментів у режимі онлайн, яка включає в себе інтерфейс користувача та сервер, який поділяється на три сервіси. Також, здійснено тестування даної системи на швидкодію та продуктивність системи.

В даній системі описана програмна складова розробки, а саме що таке програмна система, які технології були використані тощо. Програмна система – це група інтегрованих програмних засобів, які підтримують певний процес споживача і спільно використовують базу даних. В даній системі наведено методику роботи трьох сервісів: головного сервісу, додаткового сервісу та сервісу постачальника. Також наведені режими їхньої роботи, їхні можливості та конкретно, що вони в себе включають та являють. Наданий опис всіх технологій, які використовувались під час розробки даної системи, зі всіма перевагами та недоліками використання тих чи інших технологій та системи в загальному. На основі цієї методики реалізована програмна система для пошуку медикаментів у режимі онлайн. Запропоновані різні можливості для пошуку медикаментів, за допомогою пошуку, та за допомогою розташування аптек на Google map. Також в даній системі запропонований досить простий інтерфейс для користувача.

Було надано опис середовища розробки та певні його функції. Наведено, що таке REST API, там чому саме воно використовувалось під час розробки даної системи. Також, наведено, як запускається програма, а саме порядок дій, з їхнім описом, який пояснює як саме потрібно запустити програму, та які додаткові плагіни потрібно встановити.

В даній статті наведено алгоритми роботи програмної системи для пошуку медикаментів у режимі онлайн, наведені та описані основні функції системи, додано скріншоти основних модулів серверної частини програми, а також скріншоти баз даних. Наведено таблицю тестування, в якій проаналізована швидкодія та продуктивність даної системи.

Ключові слова: програмна система, клієнт-серверна система, база даних, сервер, клієнт, технології, JavaScript, HTML.

Вступ

Актуальність вибору даної теми полягає у тому, що питання здоров'я, на теперішній час займає перше місце у кожного, практично завжди спочатку ми шукаємо ті чи інші медикаменти в інтернеті, інформацію про них, ціни, місце знаходження. І не завжди вся інформація яка нам потрібна є присутня на якомусь одному сервісі, і через це нам потрібно шукати доповнення до того чи іншого препарату який нам потрібен. Чим і ускладнюється наша робота та витрачається лишній час.

Метою роботи було розробити систему, яка буде використовуватись для допомоги людям у пошуку тих чи інших медикаментів, з можливістю їх доставки у місце яке ви вкажете, при обробленні замовлення. А також з додаванням новин та лікарень Львова на даний сервіс. Ідея додавання новин та лікарень полягає у тому ж спрощенні роботи для користувача даного сервісу, а саме мати на одній сторінці зразу всю інформацію, що стосується здоров'я.

Сучасних розробок за даною ідеєю з однієї сторони вдосталь, але з іншої дуже мало. Якщо даний проект прирівнювати до інтернет магазину то можна навести такі сучасні розробки як:

Rozetka, Prom, OLX. Але, якщо взяти більш стисліше, то аналогом даної системи можна вважати сервіс tabletki.ua, що правда він значно менший в плані функціоналу, оскільки даний сервіс лише продає медикаменти [1].

Науково-технічна новизна даного проекту закладається у своїх можливостях, даний сервіс передбачає: пошук медикаментів, купівлю медикаментів, замовлення та доставку певних медикаментів, новини та інформація про лікарні Львова, а саме години роботи відділень, види відділень, години прийому та інформація про самі відділення, чим займають з чим звертатись тощо.

Основними положеннями даного програмного продукту вважається система для пошуку медикаментів, та сервер постачальника це основні складові на яких власне і створений даний програмний продукт.

Отже, як описано вище, дана система складається з трьох сервісів та трьох БД і якщо це все спробувати розмістити на одному сервісі, то це б значно сповільнило його роботу, оскільки сервісу потрібно би було опрацьовувати тисячі запитів у секунду, що могло б привести до значних затримок на ньому, тим самим сповільнило б його швидкодію та продуктивність. Саме для запобігання цьому було розроблено три сервіси. Друга задача яка була вирішена у даному проекті це задача спрощення роботи та забезпечення простішої роботи користувачам на даному сервісі. В чому полягала проблема та її вирішення. Проблема полягала у доданні на даний сервіс всіх складових, що стосуються здоров'я, новин, лікарень, медикаментів. Це було зроблено, для спрощення роботи користувачам та для заощадження їхнього часу, щоб їм не прийшлося шукати різноманітні сервіси для пошуку тої чи іншої інформації, оскільки все це додано сюди.

Аналіз останніх досліджень та публікацій

Програмна система – це група інтегрованих програмних засобів, які підтримують певний процес споживача і спільно використовують базу даних. В даній роботі, програмна система підрозуміває клієнт-серверну систему.

Клієнт-серверна система – це процес отримання запитів від клієнта їх оброблення, та відповіді на них, також це процес зберігання інформації яка вже є на сервері або яку клієнт надсилає на нього. Одним словом можна сказати що це процес взаємодії клієнта з сервером та навпаки сервера з клієнтом[2].

Клієнт-серверні системи можна поділити на двох-рівневі та трьох-рівневі системи. У двох-рівневих системах клієнт безпосередньо звертається до сервера, у трьох-рівневих системах запити обробляються проміжними серверами перш ніж передати запит до основного. Взаємодія клієнтського та серверного процесів виконується за допомогою програмного забезпечення пересилання даних. Ця взаємодія складається з декількох рівнів програмного забезпечення, що дозволяють обмінюватись даними між клієнтом та сервером. Всі запити і відповіді клієнту та сервера передаються по мережі як в певному роді повідомлення, ви яких міститься керуючу інформація та дані. Програмне забезпечення передачі даних між клієнтом та сервером здійснюється за допомогою так званих SQL виразів. SQL – це мова програмування для взаємодії між користувачем та Базами даних. А самі Бази даних – це сукупність певних даних та взаємозв'язки між ними[3, 4].

Клієнт-серверні системи управління базами даних повинні забезпечувати[2]:

- Вільний та прозорий доступ до даних.
- Надсилання по мережі клієнтських запитів.
- Обробку запитів.

У клієнт-серверних системах зменшується навантаження на мережу, а також надаються ресурси які є звільняються для виконання інтерфейсних функцій.

Основними компонентами програми, клієнт-серверної системи є:

1. Введення та висновок – це є логіка так званого уявлення.

2. Обробка – це є логіка так званого введення, а також, бізнес логіка та логіка управління даними.

3. Зберігання – це логіка так званого маніпулювання даними.

Щодо дворівневої клієнт-серверної архітектури. Вона передбачає взаємодію двох програмних модулів, а саме клієнтського та серверного.

Розрізняють такі моделі дворівневої клієнт-серверної архітектури[2]:

1. Модель тонкого клієнта, в даній моделі вся логіка застосунку та управління даними належить серверу, а клієнтська програма, забезпечує лише функції рівня представлення.

2. Модель товстого клієнта, а в цій моделі, обробка інформації та інтерфейс користувача належить клієнту, а серверу належить лише керування даними. До прикладів товстих клієнтів можна віднести: кишенькові комп'ютери, мобільні телефони та інші пристрої.

Також, є тривірнева клієнт-серверна архітектура. Вона передбачає відділення прикладного рівня від управління даними. В даній архітектурі відбувається відокремлення окремого програмного рівня, на якому є зосереджена прикладна логіка застосунку. А ось, що стосується програми проміжного рівня, то вони можуть функціонувати під управлінням спеціальних серверів 14 застосунків, проте запуск може здійснюватися і під час управління веб-сервера[2].

Для роботи з самою системою користувач буде використовувати звичайне(стандартне) програмне забезпечення, одним словом простий браузер. Ну і також це дозволить уникнути користувачеві певного роду важкостей, оскільки йому не прийдеться завантажувати та інстальовувати спеціальне програмне забезпечення, правда інколи, все-таки така необхідність виникає. Також користувачеві слід надати доступ до інтерфейсу, який дозволить йому взаємодіяти з системою, а також формувати запити до неї. Дані форми, які визначають даний інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит і після цього пересилає його до сервера, який вже буде здійснювати обробку. Також, при необхідності сервер може викликати серверні програмні модулі, які будуть забезпечувати обробку запитів і вразі виникнення потреби, будуть звертатися до сервера даних. Сервер даних – це сервер який здійснює операції над даними, що зберігаються в системі а також складають інформаційну основу. Дані, з якими працює сервер даних, як правило, організовані як реляційна база даних. Також слід додати, що веб-сервер і серверні модулі найчастіше проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою абсолютно окремі і логічно незалежні програмні модулі.

У [1] описано основний сценарій того, як правильно побудувати систему клієнт-сервер і як працює зв'язок у цих системах.

У [2] показано базовий сценарій створення такої системи з самого початку та показує, як правильно встановити з'єднання між клієнтом, сервером, клієнтом (панель адміністратора).

У [3] описані методи створення програми за допомогою node.js та express.js. Де Node.js – це платформа, побудована на середовищі виконання Chrome JavaScript для легкого створення швидких і масштабованих серверних програм на JavaScript або TypeScript

У [4] описані основні методи побудови SQL для зв'язку з сервером.

У [5] описані основні методи роботи з JavaScript, всі його основні концепції, та чому він такий популярний на теперішній час.

У [6] показана вся концепція того, що таке MySQL і як він працює.

У [7] показано що таке HTML5, його різниця з попередніми версіями, а також де він використовується?

У [8] описані методи створення програми за допомогою node.js та express.js. Де Node.js – це платформа, побудована на середовищі виконання Chrome JavaScript для легкого створення швидких і масштабованих серверних програм на JavaScript або TypeScript

У [9] описані методи створення програми за допомогою node.js та express.js. Де Node.js – це платформа, побудована на середовищі виконання Chrome JavaScript для легкого створення швидких і масштабованих серверних програм на JavaScript або TypeScript

У [10] описано що таке REST API, так як створювати подібного роду запити за допомогою HTTP.

У [11] описано API Google Maps, що це таке, для чого воно використовується, та як його підключити до власного застосунку.

У [12] показано основні методи node.js і кількість потоків у node.js, що таке цикл подій, як створити просту службу за допомогою node.js і як спілкуватися з клієнтом за допомогою node.js

Метою статті є методика побудови програмної системи для пошуку медикаментів у режимі онлайн, та тестування даної системи на продуктивність та швидкодію. Швидкодія в свою чергу не повинна бути більшою за 2,7 секунди, в той час як продуктивність не повинна бути більшою за 5,5 секунд. На основі мети, була розроблена та протестована дана система. Дана система вважається актуальною якщо дані показники буде досягнуто.

Алгоритм створення програмної системи

Основна ідея створення даної системи полягала в її унікальності, нище в таблиці 1 наведено порівняння з системою аналог tabletki.ua.

Таблиця 1

Результати порівняння систем аналогів

Програмна система для пошуку медикаментів у режимі онлайн	Tabletki.ua
Пошук, купівля та доставка медикаментів у різні райони міста	Пошук, купівля медикаментів та замовлення на самовиніс
Пошук аптек на карті, та зв'язок з ними	Пошук аптек на карті, та зв'язок з ними
Підключення бази даних лікарень Львова, з інформацією про всі відділення, години роботи, надана інформація про всіх лікарів з годинами їхнього прийому тощо.	-
Підключення API новин які стосуються ліків, стану вірусів на теперішній час, одним словом всіх новин, що стосуються здоров'я у режимі онлайн	-

Отже, як ми бачимо на таблиці 1, дана система має певні переваги на системою аналог, при чому у теперішній час ці переваги є суттєвими, оскільки це значно спрощує роботу, та заощаджує час користувачу, оскільки все це знаходиться на одному веб-сервісі.

Створюючи власну програмну систему, необхідно було детально дослідити технології які можна використовувати при їх розробці. Ознайомитись із найпопулярнішими мовами програмування, бібліотеками та інструментами для проектування програмної системи для пошуку медикаментів у режимі онлайн. На основі їх аналізу обрати найоптимальніші засоби розроблення. Для розроблення даної системи була використана технологія REST. REST-це стиль архітектури API для розподілених систем. Він визначає, які дані надаються клієнту в зручному для нього форматі. Обмін даними відбувається у форматі JSON або XML.[10] Використання REST API у розробці є дуже доцільним, адже базовою мовою програмування є мова JavaScript [1, 5]. За допомогою REST можна створити власну клієнт-серверну систему, в якій швидкодія системи буде меншою за 5,5 с, та не буде перевищувати 2,7 с.

На основі проведеного аналізу та досліджень, для розроблення користувацького інтерфейсу вибираємо:

1. Java Script, яка ідеально підходить для створення такого типу систем
2. Node.js для серверної частини;
3. Express.js фреймворк;
4. MySQL реляційна база даних.

Для реалізації проекту використовуємо технологію REST API, який на всіх рівнях стеку використовує JavaScript. Основні обмеження виглядають так:

- Клієнт-серверна архітектура;
- Взаємодія без збереження стану;
- Логічний інтерфейс додатка.

Розглянемо алгоритм створення програмної системи для пошуку медикаментів у режимі онлайн.

1. Встановлення програмного забезпечення Node.js.
2. Створення серверної частини додатка.
 - Налаштування сервера та залежностей.
 - Розроблення програмного забезпечення сервера та його запуск.
 - Створення бази даних.
 - Організація маршрутів.
 - Додавання об'єктів до бази даних та тестування сервера
3. Створення користувацького інтерфейсу:
 - 1) запуск проекту;
 - 2) встановлення додаткових залежностей та структура даних;
 - 3) Запити з frontend в backend;
 - 4) результати створення користувацького інтерфейсу.

Як серверну технологію використовуємо Node.js, який потрібно встановити за посиланням <https://nodejs.org/> [3, 8, 12]. Після встановлення Node.js рекомендовано переконавшись в успішному встановленні та перевірити їх версії.

Створення серверної частини додатка

Архітектуру серверної частини показано на рис. 1. Налаштовують сервер та залежності так.

Сервер створюємо за допомогою технології Express.js – це HTTP модуль Node.js, для створення повнофункціонального REST API. Він потребує дуже мало коду, для встановлення пакетів express використовуємо команду `npm i -save express`.

Для того, щоб встановити всі необхідні залежності, використовуємо команду `npm install express body- parser cors mongoose nodemon`, яка означає таке:

1. Express – серверний фреймворк;
2. Nodemon – перезавантажує сервер коли він побачить зміни;
3. Cors – пакет проміжного програмного забезпечення;
4. Mysql2 – для моделювання об'єктів mySQL для Node.js

На рисунку 1, зображено архітектуру програмної системи для пошуку медикаментів у режимі онлайн, а також всі зв'язки взаємодії в даній системі.

Встановивши всі необхідні залежності, можна створювати Node.js файл, який запустить сервер на порті 8080. На цьому етапі розробки створюється серверне програмне забезпечення, командою `npm start` запускається сервер.

Створення бази даних

Для створення бази даних використовуємо реляційну базу даних mySql. Для початку роботи необхідно встановити workbench за таким посиланням: <https://mysql.com/products/workbench> [4, 6]. Після успішного встановлення необхідно запустити workbench натиснувши на нього. Наступним етапом створюємо базу даних з відповідною назвою. Після успішного створення бази даних її необхідно поєднати з сервером, для чого створюється дерикторія db/, в якій міститиметься база даних, а за допомогою mongoose бібліотеки встановлюємо з'єднання. Об'єкт лікарень для бази даних матиме такий вигляд. Цей об'єкт є шаблонним об'єктом. Створення маршрутів та операції CRUD та кінцеві точки REST має такий вигляд. Для тестування та додавання об'єктів до бази даних

використовуємо інструмент MySQL Workbench. Для тестування необхідно додати кілька об'єктів до бази даних. Варто наголосити, що об'єкти створюються в форматі JSON, рис. 2.

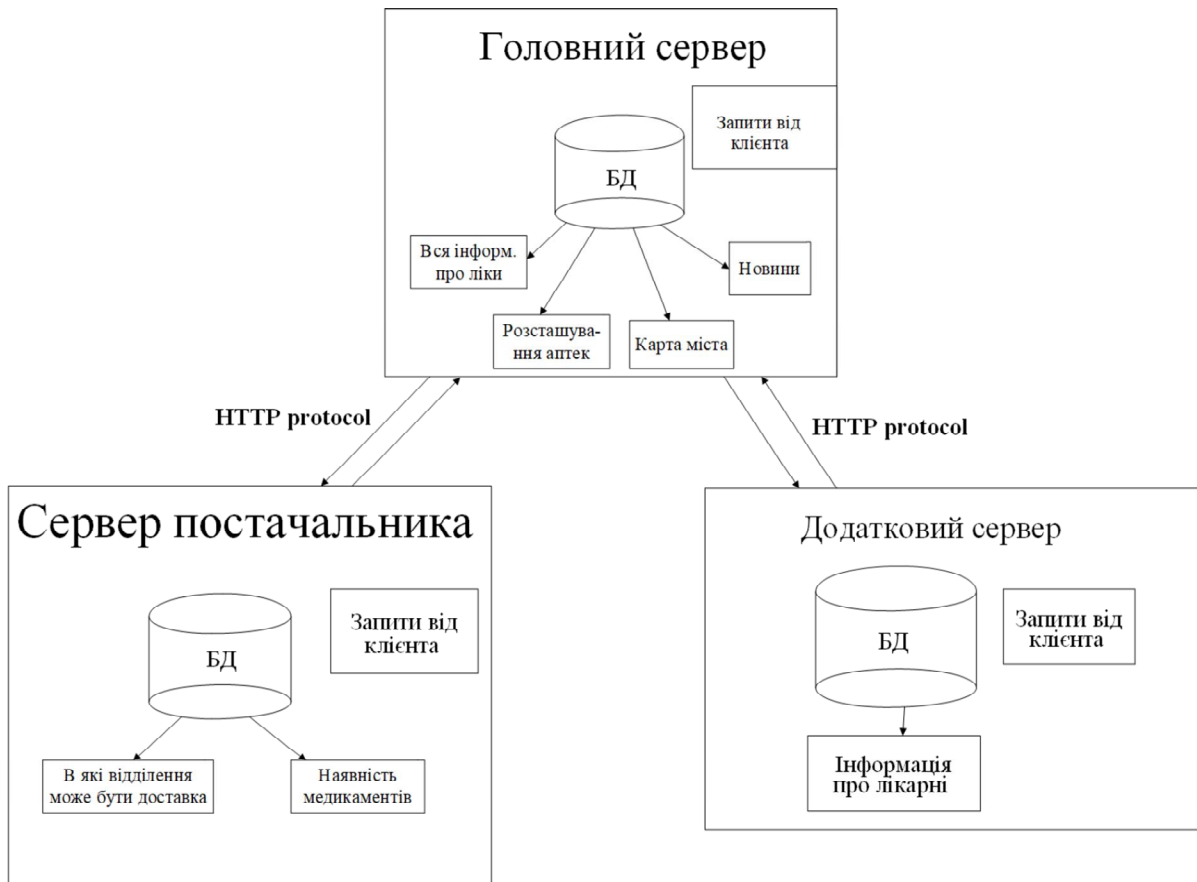


Рис. 1. Архітектура серверної частини

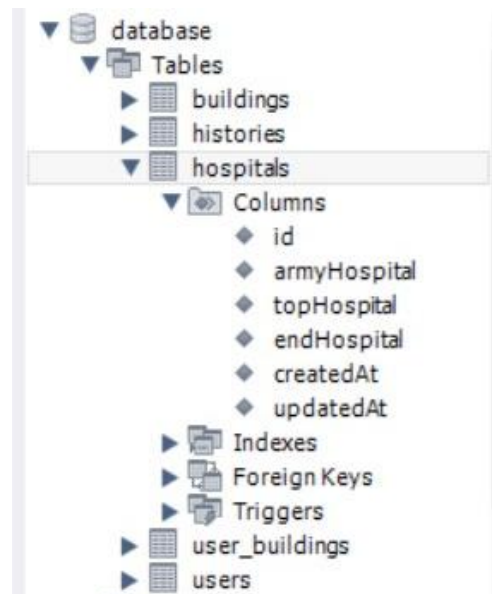


Рис. 2. База даних лікарень

На рисунку 2 зображено базу даних лікарень Львова, тут вказано структуру бази даних. Методологія запитів побудована на базі CRUD – create read update delete.

5. Тестування

Таблиця 2

Результати тестування системи

Сервіси	Швидкодія	Продуктивність
Головний сервіс	3 сек	5,4 сек
Сервіс постачальника	2 сек	4,8 сек
Додатковий сервіс	2,6 сек	5,1 сек
Середні показники системи	2,53 сек	5,1 сек

В таблиці 2 наведено результати тестування даної системи, а саме трьох головних сервісів. Якщо порівняти результати тестування, то ми бачимо, що найшвидшим сервісом є “Сервіс постачальника”, проте якщо взяти середні показники системи то ми бачимо, що середня швидкодія системи становить: 2,53 секунди, а середня продуктивність: 5,1 секунду.

6. Створення користувацького інтерфейсу та запуск проекту

Для налаштування середовища розробки використовуємо середовище Visual Studio Code, для запуску використовуємо утиліту даного середовища “Open with live server”.

Перевагою live server є те, що він не опрацьовує бекенд-логіку чи логіку баз даних, а лише надає команди для користувацького інтерфейсу, тому це можна сміло використовувати з будь-якою серверною технологією. Після успішного запуску сервера, в браузері запускається веб-сторінка, і з яким тепер можна працювати та будувати його згідно із задуманою архітектурою.



Рис. 3. Архітектура клієнтської частини

На рис. 3 зображена архітектура клієнтської частини, та показано, що саме вона в себе включає.

Також при розробці даної системи було використано Google Maps API, для того, щоб користувач міг переглядати розташування аптек в яких є ті чи інші медикаменти на карті, з годинами їх роботи.

На рисунку 4, наведено скріншот роботи Google Maps API, та наведено приклад, як розташовуються аптеки у місті Львів.

Алгоритми роботи програмної системи для пошуку медикаментів у режимі онлайн наведено на рисунку 5.

Результатом створення інтерфейсу користувача є зручні, прості та доступні веб-сторінки. У даній системі існує головна сторінка, сторінка пошуку медикаментів, сторінка лікарень Львова, сторінка постачальника та сторінка новин.

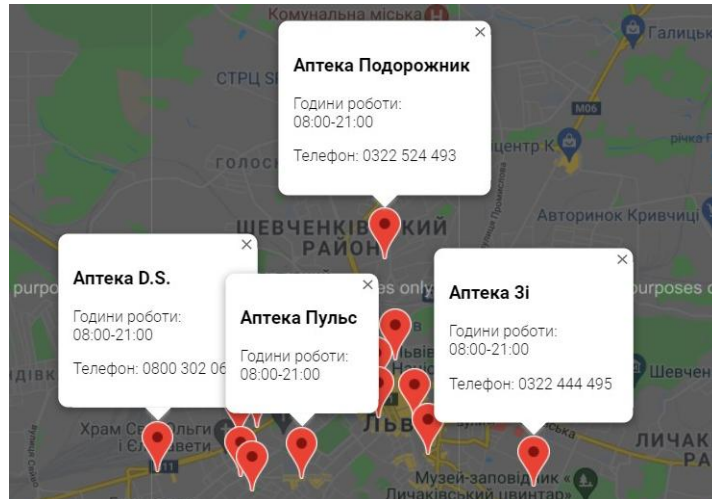


Рис. 4. Розташування аптек на Google Maps API

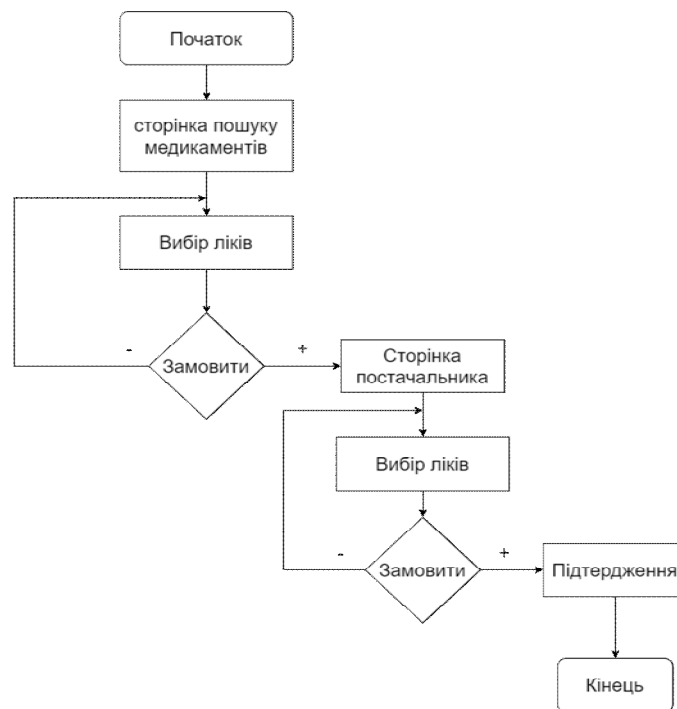


Рис. 5. Алгоритм роботи пошуку медикаментів

Висновки

Розглянуто принцип побудови програмної системи для пошуку медикаментів у режимі онлайн. Цей принцип полягає в використанні REST API. Особливість цієї системи полягає в тому, що питання здоров'я, на теперішній час займає перше місце у кожного. Ну і практично завжди спочатку ми шукаємо ті чи інші медикаменти в інтернеті, інформацію про них, ціни, місце знаходження. І не завжди вся інформація яка нам потрібна є присутня на якомусь одному сервісі, і через це нам потрібно шукати доповнення до того чи іншого препарату який нам потрібен. Програмна система для пошуку медикаментів у режимі онлайн, спрямована на спрощення пошуку тих чи інших медикаментів/інформації, та заощадження часу користувачів. Ще однією особливістю даної системи є її швидкодія та продуктивність системи. Також, у статті поетапно досліджено процес створення власної системи за допомогою REST API. З використанням JavaScript, HTML як користувацького інтерфейсу, та Node.js фреймворку Express.js для серверної розробки та MySQL

для бази даних. Результат тестування швидкості роботи створеного багатосторінкового веб-додатка: швидкодія системи показала 2,53 секунд, в той час як продуктивність 5,1 секунди. Отже виходячи із мети роботи, та порівнюючи вихідні дані з вихідними, можна сказати, що завдання яке було потрібно виконати, успішно виконано, що означає, що дана система актуальною та кращою за аналоги у даній сфері.

1. *Informatics: book of V. V. Trofimova. — M.: Publishing house Yurayt ID Yurayt, 2011. -- 911 p. Access-mode: <https://avidreaders.ru/book/informatika-v-2-t-tom-1.html>. (Accessed: 18 November 2021).*
2. *Sawsan, A. H. (2020). What is Client-Server System, New Jersey: Wiley, 5 p. DOI: 10.5281/zenodo.3673071.*
3. *Young A., Meck B., Cantelon M., Node.js in Action / Young A., Meck B., Cantelon M. Manning, 2018. 432 p. Access mode - <https://livebook.manning.com/book/node-js-in-action-second-edition/chapter-1>. (Accessed: 18 November 2021).*
4. *SQL. tproger. [Electronic resource]. – Access mode: <https://tproger.ru/translations/sql-recap/>. (Accessed: 16 November 2021).*
5. *Site of JavaScript. – Access mode <https://developer.mozilla.org/ru/docs/Web/JavaScript>. (Accessed: 16 November 2021)*
6. *Official site of MySQL. – Access mode <https://mysql.com/>. (Accessed: 16 November 2021).*
7. *HTML. Wikipedia. Free encyclopedia. [Electronic resource]. - Access mode: <https://htmlbook.ru/html>. (Accessed: 16 November 2021).*
8. *Node js, Express.js. – Access mode <https://metatnit.com/>. (Accessed: 16 November 2021)*
9. *The official website of Node.JS. Access mode <https://nodejs.org/en/>. (Accessed: 16 November 2021)/*
10. *REST API. [Electronic resource]. – Access mode: <https://sebweo.com/rozuminnya-osnov-roboti-api-i-rest-api-korotkij-vstup/>. (Accessed: 16 November 2021).*
11. *Google Maps API. [Electronic resource]. – Access mode: <https://developers.google.com/maps>. (Accessed: 16 November 2021).*
12. *Node Js. [Electronic resource].- Access mode: <https://nodejs.org/uk>. (Accessed: 16 November 2021).*

SOFTWARE SYSTEM FOR DRUG SEARCH IN ONLINE MODE

Y. Horbachov, I. Pasternak

Lviv Polytechnic National University,
Computer Engineering Department
Authors' e-mail: yura9875@ukr.net, pas_irusj@ukr.net

© Horbachov Y., Pasternak I., 2021

In this project was shown software system for drug search in online mode, which was included: three services, user interface. Also, was analyzed test of this system, speed for system and efficiency of system.

A software system – this is a group of integrated software tools which support certain process and share a database. This system shows how three services work: the primary service, the additional service and provider's service. Also given are the modes of their work, their capabilities and specifically what they include and represent. The description of all technologies are given which used during development of this system, including their advantages and disadvantages in general. On the basis of this technique, the client-server system of drug search is realized. Moreover there are various options for searching for medical items, a searching by using search and by using location of the pharmacy on Google map were suggested.

A description of the development environment and main of its functions were provided. In addition, what is REST API, where it was used and why exactly REST was used during development of this system. It also shows how to run the program, namely the procedure, with description, explains exactly how to run the program, and what additional plugins need to be installed.

This article describes the algorithm of software system for drug search in online mode, the main functions of the system are given and described. Screenshots of the main modules of the server side and of the program are added as well as screenshots of databases.

Keywords: software system, client-server system, database, server, client, technologies, JavaScript, Html.